

HOW IT WORKS

Email

Email is the most commonly used form of communication in the modern world. Simon Handby explains how your messages get from A to B

Computing breakthroughs often result in subtle and unforeseen changes to the way we live our lives. Few innovations, however, have had an impact as significant as the invention and mainstream adoption of email.

From the early messages sent between users of shared mainframes in the mid-1960s, email has grown with the availability of networked computers. Its use expanded among businesses – mostly large technology and financial corporations – during the 1970s and 1980s, but the number of messages soared as mainstream home users began to connect to the internet in the mid-1990s.

Today, email is an essential tool for governments, businesses and for millions of home users. Sadly, it's also much abused, with huge volumes of messages that contain spam and phishing attempts, and other security threats such as worms and viruses.

The scale of the internet makes it hard to establish exactly how many people have an account and how many messages are sent each day, but Microsoft says that its web-based Hotmail service alone has in the region of 280 million active accounts, which is about four per cent of the world's population. Around 100 billion emails are sent worldwide each day.

With traffic on such a scale, it's perhaps unsurprising that the basics of sending an email are so simple: you can send a plain text message from a DOS prompt, for example. Sending a mail between two accounts usually involves three steps, beginning when you compose an email.

MIME ACT

Although emails started out as simple text messages, these days they are increasingly composed in formatted text standards such

as rich text or HTML. It has also become common for people to attach pictures or other files, resulting in far bigger and more complex messages than mail systems were initially designed for.

The first internet communications used seven-bit data, with the eighth bit reserved for parity (error checking). A seven-bit number has only 127 possible values, and was used in early emails to represent the American Standard Code for Information Interchange (ASCII) character set. Virtually all internet traffic is now handled by so-called 'eight-bit clean' systems, but email software must still produce seven-bit messages to ensure successful transmissions.

To transmit an email written with extended character sets containing international characters and symbols such as the euro symbol, it's necessary to encode it into seven bits. This encoding is defined in the Multipurpose Internet Mail Extensions (MIME), which also provide for multipart messages and the encoding of binary files (such as pictures) into the body of the email. Most email clients use MIME to encode an HTML email into two sections, adding an unformatted plain text version for compatibility with text-only readers.

Because the eighth bit of every byte can't be used for data, an encoded email is usually a little larger than the information it contains.

FIRST SERVICE

For an email to be transmitted it is first sent from the client to a local email server. Large organisations tend to use server software such as Microsoft's Exchange or IBM's Lotus Notes, where the email client remains permanently connected to the server.

For home users and smaller networks, though, simple clients such as Outlook Express

or Mozilla Thunderbird are used. These talk only to the local email server when a message needs sending, and communicate using the Simple Mail Transfer Protocol (SMTP). This is the typical setup for a home user who sends email via their internet service provider's (ISP's) email server. As the name suggests, SMTP is a text-based protocol that you can even use to send a message directly (see the 'First-class post' box).

To prevent unauthorised users from sending email, a mail server may require authentication before it will accept SMTP traffic destined for users based on another server. In the case of a home user sending via their ISP, the server will usually recognise that the mail originates from the ISP's internal network and allow it.

This is a near-foolproof system, but it may mean that the same computer used elsewhere – a notebook connecting via a public hotspot, for example – will be unable to send mail. A more flexible alternative is to require a username and password each time a client connects.

It is possible to bypass your ISP or organisation's mail server altogether and send an email directly to the recipient's domain. Doing so may take longer, however, and there's no guarantee that the destination network will be ready. It's easier and quicker to upload messages to a server on your own network or at the other end of a broadband connection, and to let it take over responsibility for delivering them.

STATE YOUR DESTINATION

Once an email server has accepted a message, it will try to send it on your behalf to the mail server on the destination domain. This usually happens immediately, but in remote locations with an intermittent connection to the internet, the server may queue several messages before connecting and sending them.

First, the server must find the correct destination server. It extracts the recipient's domain from their email address, and uses DNS to find the IP address of the server that handles that domain's email. In the early days of email the sending server may have relayed the message through an intermediate server. This practice has now virtually disappeared, and all but a tiny number of emails are delivered to the destination server in one 'hop'.

Once it has found the correct destination server, the originating server contacts it using SMTP. The originator tells the destination the domain on behalf of which it is sending mail, and provides the sender's and the recipient's email addresses. The destination server will usually check that an account exists in the recipient's name before allowing the originator to continue.

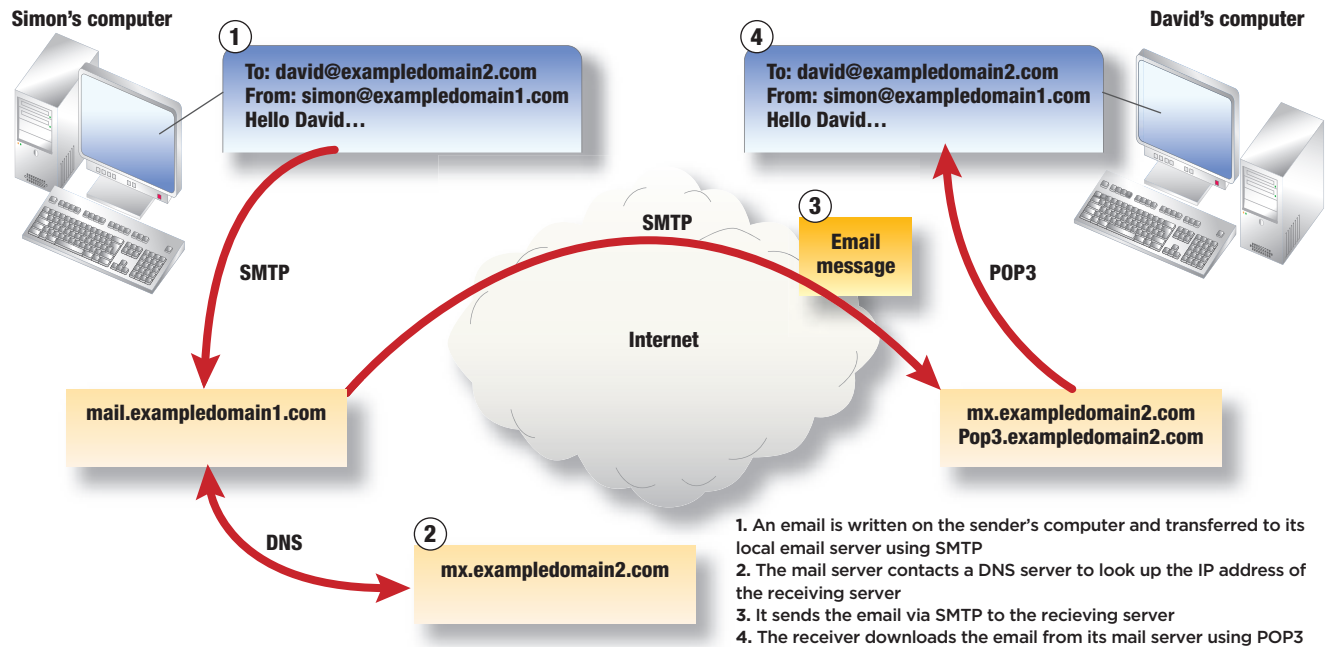
Provided there are no hitches, the originating server can now send the email's headers, which

Self service Configuring an email server

It's surprisingly easy to set up an email server. You'll need a domain that you own, an always-on internet connection such as broadband, a decent firewall and a computer that remains switched on. You'll also need a static IP address from your ISP, for which it may charge. Check that you won't be breaking its terms and conditions, too.

- Download and install some simple email server software – some, such as MailEnable, is free. Follow the publisher's configuration instructions. Only allow the server to send mail for domains you own. You can improve the server's security by requiring a password to send mail.
- Add an 'A type' DNS record for mail.yourdomain.co.uk that points to the static IP address your ISP allocates. This will direct your domain's email traffic to your router's external address.
- Add a service rule in your router or firewall that directs inbound SMTP traffic to your mail server's IP address on your internal network (typically 192.168.xxx.xxx). You may also need to give the server a static address.
- Test your server with mails to and from another account, such as a Google Mail address.
- You can troubleshoot problems with the tools available on www.mxtoolbox.com. Be sure to check that your server isn't an open relay.

HOW AN EMAIL IS TRANSMITTED



usually contain at least email addresses for the 'from' and 'to' fields and a subject. Although these email addresses are generated by the email client and are displayed to the recipient, they aren't used by either server, and need not match the addresses of the actual sender and recipient. This is a quirk that leaves the system open to abuse, as spammers can spoof email addresses and make it look as though they're coming from a different source.

Once the headers are transmitted, the remainder of the message text or MIME-encoded content is sent. The originating server will send any other messages it has for the destination domain and disconnect the session.

SECOND SERVICE

SMTP is a push protocol, where messages are sent to an ever-vigilant server, but there's no guarantee that the message recipient will be using email or even their computer when a message arrives. For this reason, incoming emails are stored in a user's server mailbox until requested – or pulled – by the mail client.

Standard email clients normally achieve this through the aptly named Post Office Protocol (POP3), which, like SMTP, is based on simple text commands. By default, email clients using POP3 check every few minutes for new mail, providing a username and password before the server will allow them to download and delete messages.

It's easy to configure POP3 clients to leave a copy of messages on the server, but POP3 works on the assumption that mails will be downloaded and stored elsewhere. Some mail servers support the more sophisticated Internet Message Access Protocol (IMAP), however, which works either in an offline mode similar to POP3, or in an online mode where the user remains connected to the server.

IMAP considers the user's mailbox on the server to be the authoritative message store for an account. Messages can be downloaded to the client, but to delete them you must explicitly delete them from the server. IMAP supports the download of individual parts of MIME-encoded messages, and also allows server-side searches. Although the extra functionality can be a

great advantage to users with a slow connection, IMAP servers require more storage space and must have the resources to perform more advanced jobs.

A SPAMMER IN THE WORKS

The protocols used by email evolved at a time when messages were relayed from origin to destination via servers linked by dial-up connections. Because long-distance calls were expensive, mails were routed or relayed through many servers, which were generally configured to accept mail from any user for any destination (an 'open relay'). Setting up a mail server wasn't quite as straightforward as it is now (see the 'Self service' box opposite) and abuse of the system wasn't a major concern.

Modern mail servers are permanently connected to the internet, however. As the number of email users began to soar in the 1990s, unsolicited marketing mails became more common. Spammers soon realised that they could use somebody else's email server to relay messages, using that server's resources and connection bandwidth to reach more inboxes. To combat this, and because relaying had become redundant for email traffic, mail administrators began to stop their servers relaying emails for unauthenticated senders.

These days open relays are rare, and their vulnerability to exploitation means that they are quickly blacklisted by anti-spam services. With high-bandwidth, always-on internet now widespread among home users, spammers send the bulk of their messages from distributed networks of computers infected by worms. The number of spam emails has rocketed, and they are now thought to make up around 80 per cent of all email – up to 90 billion messages per day.

Aside from the human frustration of having unsolicited and often offensive emails arrive in your inbox, spam clogs the internet with massive, wasteful traffic. As anti-spam programs have become better at automatically filtering out bulk messages, spammers have begun to attach less easily analysed JPEG or GIF image files to convey their message, resulting in bigger email sizes and yet more traffic. **CS**

First-class post

Sending a simple email

Not all email servers are configured to allow anyone to send a message, but you can connect to one that is from a DOS prompt by typing 'telnet www.exampledomain.com 25'. If the server response looks similar to the first line below, you may be able to send an email by typing the commands beginning >: Remember to substitute real addresses for the examples we've used:

```
220 mail.exampledomain1.com ESMTP
Postfix ready
>: HELO exampledomain2.com
250 Hello exampledomain2.com
>: MAIL FROM:simon@exampledomain2.com
250 Ok
>: RCPT TO:editor@exampledomain1.com
250 Ok
>: DATA
354 End data with <CR><LF>.<CR><LF>
>: Subject: How email works
>: From: simon@exampledomain2.com
>: To: editor@exampledomain1.com
>:
>: Hello David,
>: If you can read this the email article is going well!
>: Simon.
>:
250 Ok: queued
>: QUIT
221 Bye
```

FURTHER INFORMATION

Wikipedia entry on email
<http://en.wikipedia.org/wiki/Email>

Monash University email statistics – daily traffic and spam figures for a 60,000-student university
www.tinyurl.com/2wzkhd